

This Page Is Inserted by IFW Operations  
and is not a part of the Official Record

## **BEST AVAILABLE IMAGES**

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images may include (but are not limited to):

- BLACK BORDERS
- TEXT CUT OFF AT TOP, BOTTOM OR SIDES
- FADED TEXT
- ILLEGIBLE TEXT
- SKEWED/SLANTED IMAGES
- COLORED PHOTOS
- BLACK OR VERY BLACK AND WHITE DARK PHOTOS
- GRAY SCALE DOCUMENTS

**IMAGES ARE BEST AVAILABLE COPY.**

**As rescanning documents *will not* correct images,  
please do not report the images to the  
Image Problem Mailbox.**



## PATENT ABSTRACTS OF JAPAN

(11) Publication number: **06052182 A**(43) Date of publication of application: **25.02.94**

(51) Int. Cl.

**G06F 15/21**(21) Application number: **04202386**(22) Date of filing: **29.07.92**(71) Applicant: **TOSHIBA CORP**(72) Inventor:  
**OWA TSUTOMU  
ISHII AKIRA  
ABE TETSUYA**(54) **ASSIGNING DEVICE**

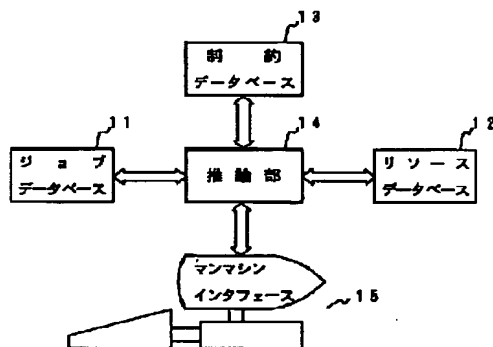
the constraint data base 13.

(57) Abstract:

COPYRIGHT: (C)1994,JPO&amp;Japio

**PURPOSE:** To facilitate the decision of good assignment suitable for a resource by setting a more detailed constraint condition based on circumstance the resource requires in accordance with the degree of that constraint.

**CONSTITUTION:** This assigning device is provided with, at least, a job data base 11 which stores and preserves plural jobs with the attribute of starting time and ending time, a resource data base 12 which stores and preserves the plural resources, a constraint data base 13 which stores and preserves respectively the constraint condition at an absolute level at the time of assigning the job to the resource and the constraint condition at a normal level to be a standard of the starting time and the ending time, an inferring part 14 which takes out the job preserved in the job data base 11 in the order of its time and assigns it to the resource preserved in the resource data base 12 in conformity with the constraint condition at the absolute level and the constraint condition at the normal level preserved in the constraint data base 13, and a man-machine interface 15 which rewrites and changes the contents of the constraint condition preserved in



(19)日本国特許庁(JP)

(12) 公開特許公報(A)

(11)特許出願公開番号

特開平6-52182

(43)公開日 平成6年(1994)2月25日

(51)IntCl.<sup>5</sup>

G 0 6 F 15/21

識別記号

庁内整理番号

F I

技術表示箇所

Z 7052-5L

審査請求 未請求 請求項の数1(全 13 頁)

(21)出願番号 特願平4-202386

(22)出願日 平成4年(1992)7月29日

(71)出願人 00003078

株式会社東芝

神奈川県川崎市幸区堀川町72番地

(72)発明者 大輪 勤

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

(72)発明者 石井 暁

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

(72)発明者 阿部 哲也

神奈川県川崎市幸区柳町70番地 株式会社

東芝柳町工場内

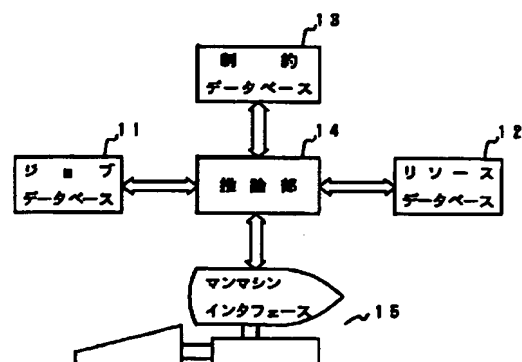
(74)代理人 弁理士 鈴江 武彦

(54)【発明の名称】 割当て装置

(57)【要約】

【目的】 リソースの要求する事情に即してより詳細な制約条件をその制約の度合いに応じて設定し、リソースに適した優れた割当て決定を容易に行なう。

【構成】 少なくとも開始時刻及び終了時刻の属性を有するジョブを複数記憶保存するジョブデータベース11と、リソースを複数記憶保存するリソースデータベース12と、ジョブをリソースに割当て際の絶対的レベルでの制約条件と開始時刻及び終了時刻の目安となる標準的レベルでの制約条件とをそれぞれ記憶保存する制約データベース13と、上記ジョブデータベース11に保存されるジョブをその時刻の順に取出して上記制約データベース13に保存される絶対的レベルでの制約条件及び標準的レベルでの制約条件に従い、上記リソースデータベース12に保存されるリソースに割当てする推論部14と、上記制約データベース13に保存される制約条件の内容を書換変更するマンマシンインタフェース15とを備える。



## 【特許請求の範囲】

【請求項1】 少なくとも開始時刻及び終了時刻の属性を有するジョブを複数記憶保存するジョブデータベース部と、

リソースを複数記憶保存するリソースデータベース部と、

ジョブをリソースに割当てる際の絶対的レベルでの制約条件と開始時刻及び終了時刻の目安となる標準的レベルでの制約条件とをそれぞれ記憶保存する制約データベース部と、

上記ジョブデータベース部に保存されるジョブをその時刻の順に取出して上記制約データベース部に保存される絶対的レベルでの制約条件及び標準的レベルでの制約条件に従い、上記リソースデータベース部に保存されるリソースに割当てる推論部と、

上記制約データベース部に保存される制約条件の内容を書換変更する変更部とを具備したことを特徴とする割当て装置。

## 【発明の詳細な説明】

## 【0001】

【産業上の利用分野】 本発明は、複数のジョブと複数のリソースとの間の組合わせを効果的に割当て決定する割当て装置に関する。

## 【0002】

【従来の技術】 従来より、複数のジョブと複数のリソースとの間の組合わせを、種々の制約条件の下で割当て決定することが多く行われてきた。例えば、種々の業務（ジョブ）に対する作業者の割当てを、労働条件や作業能力（制約条件）に従ってそれぞれ割当て決定することが行なわれる。

【0003】 このような割当て決定は、従来、専ら人手に頼って行なわれているが、上記ジョブとリソースとがそれぞれ大量になるに従って人手による割当て決定が困難化する。また、割当て決定を進める過程で、例えばあるジョブに割当て得るリソースが既に他のジョブに割当て決定されている結果、そのジョブに割当てリソースがなくなる等の矛盾が生じ易く、割当て決定処理の行き詰まりが生じることも多々ある。そして、この場合には、その割当て決定手続きをやり直す必要がある。このため、ジョブとリソースとの割当て決定に多大な時間を必要とすることが否めなかった。

【0004】 そこで最近では、この種の割当て決定を計算機を用いて処理することが考えられている。この計算機による処理は、基本的に複数のジョブの中から1つのジョブを取出し、取出されたジョブに割当て可能なリソースを、複数のリソースの中から順に取出されるリソースについてその割当て制約条件等をチェックして求める。このチェックは、具体的には割当ての制約条件をチェックした後、さらにその動的条件をチェックすることによって行なわれる。そして、割当て制約条件等を満た

すリソースの中から、例えば最初に見出だされたリソースを上記ジョブに割当て決定し、ジョブに対して割当て決定し得るリソースが見出だされない場合はその割当て変更を行なう。このような処理手続きを、割当て決定されたジョブ及びリソースを除去しながら繰返すことによって割当て決定が実行されるものである。

【0005】 ところがこのような処理手続きにあっては、ジョブ及びリソースの数の増大に伴って割当て決定が非常に難しくなり、処理を実行させるためのプログラムの作成も困難となる。

【0006】 そこで、ジョブの集合を格納するジョブデータベース、リソースの集合を格納するリソースデータベース及び各種制約を格納する制約データベースをそれぞれ分離独立して設け、割当て決定を行なう推論部がジョブデータベースからジョブを1つずつ取出し、制約データベースに格納されている制約を満たすようにリソースデータベースに格納されるリソースに割当てる方式が考えられている。

【0007】 しかしながら上記のようにジョブデータベース、リソースデータベース及び制約データベースを分離させ、これらデータベースの内容に基づいて推論部が順次割当て決定する方式では、ただ単に制約データベースに格納された禁止要項等の絶対的な制約条件等のみによって割当て決定を行なうため、必ずしも優れた割当て決定が得られるとは限らず、例えばジョブが作業でリソースが作業者であるとする、作業者にとって甚だ作業がしづらいような割当て決定がなされることもあり得る。

## 【0008】

【発明が解決しようとする課題】 このように従来にあっては、割当て決定の基準となる制約条件として禁止要項等の絶対的なもののみを使用して割当て決定を行っていたため、不可ではないがリソースにとってはあまり具合のよくない割当て決定がなされてしまう可能性がある。

【0009】 本発明は上記のような実情に鑑みてなされたもので、その目的とするところは、リソースの要求する事情に即してより詳細な制約条件を設定し、リソースに適した優れた割当て決定を容易に行なうことが可能な割当て装置を提供することにある。

## 【0010】

【課題を解決するための手段】 すなわち本発明は、少なくとも開始時刻及び終了時刻の属性を有するジョブを複数記憶保存するジョブデータベースと、リソースを複数記憶保存するリソースデータベースと、ジョブをリソースに割当てる際の絶対的レベルでの制約条件と開始時刻及び終了時刻の目安となる標準的レベルでの制約条件とをそれぞれ記憶保存する制約データベースと、上記ジョブデータベースに保存されるジョブをその時刻の順に取出して上記制約データベースに保存される絶対的レベルでの制約条件及び標準的レベルでの制約条件に従い、上

記リソースデータベースに保存されるリソースに割当てる推論部と、上記制約データベースに保存される制約条件の内容を書換変更するマンマシンインタフェースとを備えるようにしたものである。

【0011】

【作用】上記のような構成としたことにより、リソースの要求する事情に即してより詳細な制約条件をその制約の度合いに応じて設定し、リソースに適した優れた割当て決定を容易に行なうことができる。

【0012】

【実施例】以下図面を参照して本発明の一実施例を説明する。

【0013】図1はその基本構成を示すものであり、11がジョブを複数格納したジョブデータベース、12がリソースを複数格納したリソースデータベース、13が複数レベルの制約条件を格納した制約データベース、14が上記制約データベース13に格納される制約条件に従ってジョブデータベース11に格納されるジョブをリソースデータベース12に格納されるリソースに割当て決定する推定部、15は利用者がジョブデータベース11、リソースデータベース12及び制約データベース13に格納されている各データの書換変更等を行なうためのマンマシンインタフェースである。

【0014】上記のような構成にあって、例えば路線バスの運行をジョブとし、これを仕業と呼ばれる仮想的な乗務員をリソースに対して割当てする場合の動作について説明する。

【0015】図2はジョブデータベース11に格納されるジョブとしての運行を例示するものであり、「運行1」～「運行13」の13の運行に関し、図示する如く「出発時刻」「出発場所」「到着時刻」及び「到着場所」の計4つの属性を有するデータをもって1つの運行とする。ここで、出発場所と到着場所とが共に「東京」となっているものは往復の運行を示している。

【0016】このような運行に対して、リソースデータベース12に「仕業1」～「仕業4」の4つの仕業がリソースとして格納されており、上記「運行1」～「運行13」をこれら「仕業1」～「仕業4」に割当て決定するものとする。上記割当て決定を行なうために制約データベース13に格納される制約条件を図3～図5に示す。

【0017】図3は各仕業の種類毎に就業規則のようにして課せられる制約を例示するものである。同図に示す如く一般にバス会社においては仕業を「午前」「午後」「日勤」「中休」「午前増務」及び「午後増務」の6種類とし、各始業に対して「最大拘束時間」「最大ハンドル時間」「連続ハンドル時間」等の項目をそれぞれ制約として課すものである。各項目の具体的な内容については「午前」「午後」のみ例示し、その他については省略する。以下、これら個々の各項目について自明なものは省略して説明する。

【0018】「最大拘束時間」はその仕業によって乗務員が拘束される時間の限度を示し、「最大ハンドル時間」は上記「最大拘束時間」中の実際にバスを運転する時間の限度を示す。また、「連続ハンドル時間」はその時間を超えて連続運転してはならない限度を示し、後述する「連続ハンドル中断場所」にて運転せず止まっていることによりそのカウントが中断される。

【0019】「開始時刻目安」「終了時刻目安」は、その仕業が勤務する時間帯の目安を示すもので、あくまでも「目安」であるので、他の項目のように絶対的なレベルでの制約ではなく、標準的なレベルのものとする。

【0020】上記のように制約条件として、必ず厳守しなければならない絶対的なレベルの項目と、「目安」と付したやや制約としての度合いの低い標準的なレベルの項目とを併せて用いる。

【0021】図4は、ある仕業を終了してから別の仕業が開始されるまでの時間の制約条件を例示する図であり、1台のバスを仕業間で共用するような場合に関係する。例えば、「午前」と「午後」の仕業は1台のバスを連続して用いるので、その間に「10分以上」の時間が必要なが示されている。これは、具体的には、「午前」の仕業の1つのリソースについて「午後」の仕業の1つのリソースが対応し、その間に最低「10分」の時間が必要であることを示す。このように「午前」の仕業のリソースのすべてについて、「午後」の仕業のリソースが原則として対応するようになり、「午前」と「午後」の仕業のリソースの数が異なる場合には、その多い分に関しては対応付けが不要となる。

【0022】また、「午後増務」の仕業は、実際には「午前」の仕業の残業として運用されるので、その間に「120分」の間隔が必要なが表わされており、「午前増務」の仕業と「午後」の仕業の間も同様となる。また、別の制約条件として、図5に例示するようなバスの路線に関する種々項目があるので、各項目について以下に説明する。

【0023】「停留所」はバスを停車させる停留所名すべてを示し、「休憩場所」は上記図3の「休憩時間」に対応する休憩をとることができる停留所名を示す。「連続ハンドル中断場所」はそこで一定の時間以上運転せず止まっていれば連続ハンドル時間が中断される場所を示し、「最小折待時間」はその停留所に発着する運行を接続する時に運行間にとらなければならない最小の時間を示す。

【0024】次いで、上記のような各種制約条件が制約データベース13に格納された状態でジョブデータベース11に格納されたジョブとしての「運行1」～「運行13」をリソースデータベース12に格納されるリソースとしての4つの仕業「仕業1」～「仕業4」に割当て決定する際の推定部14の動作を説明する。この場合、「仕業1」及び「仕業2」は図3及び図4で示した「午前」の

型、「仕業3」及び「仕業4」は図3及び図4で示した「午後」の型であるものとする。図6は推定部14による割当て決定の基準となる知識を示すもので、その内容はそれぞれ後述する。

【0025】図7は上記図6の知識に基づいて推定部14が行なう割当て決定の処理内容であり、まず推定部14がジョブとしてジョブデータベース11に格納されている

「運行1」を時刻の早い順に取出し（ステップA1）、次いでリソースとしてリソースデータベース12に格納されている「仕業1」を記述されている順に取出す（ステップA2）。それぞれに取出した「運行1」が「仕業1」に接続可能であるか否かを判断し（ステップA3）、可能であると判断すると、次に割当てたものとして制約を満たすか否かを制約データベース13に格納されている各制約条件を基準に判断する（ステップA4）。これは、図6に示すような「知識1・ジョブが接続可能で、かつ接続した時、制約を満たすリソースがあるなら、該リソース中の記述順序の早いリソースに割当てる。」なる知識に基づいての処理であり、「運行1」は上記図3～図5の各制約条件を満たすため、そのままこの「運行1」をリソースとして「仕業1」に割当てる（ステップA5）。割当てを終えた後、割当てをすでに終えてそれ以上の割当てを行なう必要がないリソースに対する終了処理を行なう。

【0026】図8はこのリソースの終了処理の詳細を示し、上記図6に示す「知識4・時刻順に取出される次のジョブの開始時刻が、（リソースの）終了時刻目安を超えている時、該当するリソースへの割当てを終了する。」なる知識に基づいての処理である。処理当初にはまず割当て中のリソースである「仕業1」をリソースデータベース12から取込む（ステップB1）。その後、次に選択される筈のジョブである「運行2」をジョブデータベース11から取込み（ステップB2）、「運行2」の開始時刻が「仕業1」の終了時刻目安を超えているか否かを判断する（ステップB3）。ここで超えていないと判断すると、この「仕業1」に関しては終了させるべきものではないとして、次いでまだ割当てが終了していないリソースがあるか否かを判断する（ステップB5）。ここではまだ「仕業2」～「仕業4」に関しては割当てが終了していないため、再び上記ステップB1からの処理を繰返す。

【0027】以下、この図8の処理を順次「仕業2」～「仕業4」に関しても実行し、「仕業1」と同様に終了させるべきものではないと判断して以上でこの図8の処理を終了する。

【0028】リソースの終了処理を終えた時点で図7では次にまだ割当てを行なっていないジョブがあるか否かを判断する（ステップA13）。これは、上記図6に示す「知識5・すべてのジョブが割当てもしくは割当不可とされた時、割当て中であるリソースへの割当てを終了す

る。」なる知識に基づいての処理であり、まだジョブがあると判断すると再び上記ステップA1からの処理を繰返す。

【0029】時刻の早い順に「運行2」をジョブデータベース11より取出し（ステップA1）、次いで「仕業1」を取出す（ステップA2）。それぞれに取出した「運行2」が「仕業1」に接続可能であると判断し（ステップA3）、次に割当てたものとして制約を満たすと判断すると（ステップA4）、そのままこの「運行2」をリソースとして「仕業1」に割当てる（ステップA5）。割当てを終え、再び上記図8のリソースの終了処理を行ない、「仕業1」～「仕業4」のいずれも終了させるべきものではないとして処理を終えた後、次にまだ割当てを行なっていないジョブがあると判断すると（ステップA13）、再度上記ステップA1からの処理を繰返す。

【0030】次いで「運行3」をジョブデータベース11より取出し（ステップA1）、「仕業1」を取出す（ステップA2）。それぞれに取出した「運行3」が「仕業1」に接続可能であると判断する（ステップA3）。次に「運行3」を「仕業1」に割当てたものとする「連続ハンドル時間」が「150分」となり、制約条件「130分」を超えてしまうこととなるので、この制約を満たさないと判断して（ステップA4）、まだ割当てを行なうリソースがあるか否かを判断する（ステップA6）。ここではまだ「仕業2」～「仕業4」が残っているのので、上記ステップA2に進んで記述されている順に次のリソースである「仕業2」を取出し、今度は「運行3」が「仕業2」に接続可能であるか否かを判断する（ステップA3）。これは、上記図6に示す「知識2・ジョブが接続可能な時、記述順序の早いリソースに割当てる。」なる知識に基づいて記述順序に従って「仕業2」を取出し、接続可能か否かを判断するもので、ここでは接続可能と判断し、次に割当てたものとして制約を満たすと判断すると（ステップA4）、この「運行3」を「仕業2」に割当てる（ステップA5）。割当てを終えた後、再び上記図8のリソースの終了処理を行なう。

【0031】図8では、割当て中の「仕業2」を取込み（ステップB1）、次に選択される筈のジョブである「運行4」をジョブデータベース11から取込み（ステップB2）、「運行4」の開始時刻が「仕業2」の終了時刻目安を超えているか否かを判断する（ステップB3）。ここで超えていないと判断すると、この「仕業2」に関しては終了させる必要はないとして、次いでまだ割当てが終了していないリソースがあるか否かを判断する（ステップB5）。ここではまだ「仕業1」、「仕業3」、「仕業4」に関しては割当てが終了していないため、再び上記ステップB1からの処理を繰返す。

【0032】以下、この図8の処理を「仕業1」「仕業3」「仕業4」に関しても実行し、「仕業2」と同様に

終了させる必要はないと判断して以上でこの図8の処理を終了する。

【0033】リソースの終了処理を終えた時点で図7では次にまだ割当てを行っていないジョブがあるか否かを判断し(ステップA13)、あると判断すると再び上記ステップA1からの処理を繰返す。

【0034】次いで「運行4」をジョブデータベース11より取出し(ステップA1)、「仕業1」を取出す(ステップA2)。それぞれに取出した「運行4」が「仕業1」に接続可能であると判断し(ステップA3)、次に割当てたものとして制約を満たすと判断する(ステップA4)。これは、「仕業1」が「運行3」に該当する間、運転を中断することとなったため、「連続ハンドル時間」の制約条件を満たすと判断されるため、そのままこの「運行4」をリソースとして「仕業1」に割当てる(ステップA5)。割当てを終え、再び上記図8のリソースの終了処理を行ない、「仕業1」～「仕業4」のいずれも終了させる必要はないとして処理を終えた後、次にまだ割当てを行っていないジョブがあると判断すると(ステップA13)、再度上記ステップA1からの処理を繰返す。

【0035】次いで「運行5」をジョブデータベース11より取出し(ステップA1)、「仕業1」を取出す(ステップA2)。取出した「運行5」が「仕業1」に接続不可であると判断し、(ステップA3)、次にまだ割当てを行なうリソースがあるか否かを判断する(ステップA6)。ここではまだ「仕業2」～「仕業4」が残っているので、上記ステップA2に進んで記述されている順に次のリソースである「仕業2」を取出し、今度は「運行5」が「仕業2」に接続可能であるか否かを判断する(ステップA3)。接続可能と判断し、次に割当てたものとして制約を満たすと判断すると(ステップA4)、この「運行5」を「仕業2」に割当てる(ステップA5)。割当てを終えた後、再び上記図8のリソースの終了処理を行なう。

【0036】図8では、上記「運行3」の時と同様、「仕業2」「仕業1」「仕業3」「仕業4」のいずれに関しても終了させる必要はないと判断してこの処理を終え、まだ割当てを行っていないジョブがあると判断すると(ステップA13)、再度上記ステップA1からの処理を繰返す。次いで、「運行6」についても上記「運行4」と同様に「仕業1」に割当てる。

【0037】次に「運行7」をジョブデータベース11より取出し(ステップA1)、「仕業1」を取出す(ステップA2)。取出した「運行7」が「仕業1」に接続可能であると判断し(ステップA3)、この「運行7」を「仕業1」に割当てたものとして「最大拘束時間」が「360分」となり、制約条件「270分」を超えてしまうこととなるので、この制約を満たさないと判断して(ステップA4)、まだ割当てを行なうリソースがある

か否かを判断する(ステップA6)。ここではまだ「仕業2」～「仕業4」が残っているので、上記ステップA2に進んで記述されている順に次のリソースである「仕業2」を取出し、今度は「運行7」が「仕業2」に接続可能であるか否かを判断する(ステップA3)。接続可能と判断し、次に割当てたものとして制約を満たすと判断すると(ステップA4)、この「運行7」を「仕業2」に割当てる(ステップA5)。割当てを終えた後、再び上記図8のリソースの終了処理を行なう。

【0038】図8では、割当て中の「仕業2」を取込み(ステップB1)、次に選択される筈のジョブである「運行8」をジョブデータベース11から取込み(ステップB2)、「運行8」の開始時刻「12:10」が「仕業2」の終了時刻目安「12:00」を超えていると判断すると(ステップB3)、この「仕業2」に関しては割当てを終了させるべきであるとしてその設定を行ない(ステップB4)、次いでまだ割当てが終了していないリソースがあるか否かを判断する(ステップB5)。ここではまだ「仕業1」、「仕業3」、「仕業4」に関しては割当てが終了していないため、再び上記ステップB1からの処理を繰返す。

【0039】しかるに、続いて「仕業1」も同様に終了時刻目安「12:00」を「運行8」の開始時刻「12:10」を超えていると判断し、この「仕業1」に関しても割当てを終了させるべきであるとしてその設定を行ない(ステップB4)、次いでまだ割当てが終了していないリソースがあるか否かを判断する(ステップB5)。ここではまだ「仕業3」、「仕業4」に関しては割当てが終了していないため、再び上記ステップB1からの処理を繰返す。

【0040】その後、「仕業3」「仕業4」に関しては割当てを終了させる必要はないと判断し、以上でこの処理を終えた後、まだ割当てを行っていないジョブがあると判断すると(ステップA13)、再度上記ステップA1からの処理を繰返す。

【0041】次いで、「運行8」をジョブデータベース11より取出し(ステップA1)、「仕業1」「仕業2」はすでに上記図8の処理で終了設定されているので「仕業3」を取出す(ステップA2)。取出した「運行8」が「仕業3」に接続可能であると判断し(ステップA3)、次に割当てたものとして制約を満たすと判断すると(ステップA4)、そのままこの「運行8」を「仕業3」に割当てる(ステップA5)。割当てを終え、再び上記図8のリソースの終了処理を行ない、すでに終了設定された「仕業1」「仕業2」を除いて「仕業3」、「仕業4」のいずれも終了させる必要はないとして処理を終えた後、次にまだ割当てを行っていないジョブがあると判断すると(ステップA13)、再度上記ステップA1からの処理を繰返す。

【0042】次に「運行9」をジョブデータベース11よ

り取出し(ステップA1)、「仕業1」「仕業2」はすでに上記図8の処理で終了設定されているので「仕業3」を取出す(ステップA2)。この「運行9」を「仕業3」に接続可能であると判断した後(ステップA3)、「運行9」を「仕業3」に割当てたものとして「最小折待時間」が「5分」となり、制約条件「10分」を下回ってしまうこととなるので、この制約を満たさないと判断して(ステップA4)、まだ割当てを行なうリソースがあるか否か判断する(ステップA6)。ここではまだ「仕業4」が残っているので、上記ステップA2に進んで「仕業4」を取出し(ステップA2)、今度は「運行9」が「仕業4」に接続可能であるか否か判断する(ステップA3)。接続可能と判断し、次に割当てたものとして制約を満たすと判断すると(ステップA4)、この「運行9」を「仕業4」に割当てる(ステップA5)。割当てを終えた後、再び上記図8のリソースの終了処理を行なう。

【0043】図8では、割当て中の「仕業4」を取込み(ステップB1)、次に選択される筈のジョブである「運行10」をジョブデータベース11から取込み(ステップB2)、「運行10」の開始時刻「13:40」が「仕業4」の終了時刻目安「22:00」を超えていないと判断すると(ステップB3)、この「仕業4」に関しては割当てを終了させる必要はないと判断して(ステップB3)、次いでまだ割当てが終了していないリソースがあるか否か判断する(ステップB5)。ここではまだ「仕業3」に関して割当てが終了していないため、再び上記ステップB1からの処理を繰返し、この「仕業3」に関しても割当てを終了させる必要はないと判断し、以上でこの処理を終えた後、まだ割当てを行っていないジョブがあると判断すると(ステップA13)、再度上記ステップA1からの処理を繰返す。

【0044】次いで「運行10」をジョブデータベース11より取出し(ステップA1)、「仕業3」を取出す(ステップA2)。この「運行10」を「仕業3」に接続可能と判断し(ステップA3)、次に割当てたものとして制約を満たすと判断すると(ステップA4)、この「運行10」を「仕業3」に割当てる(ステップA5)。割当てを終えた後、再び上記図8のリソースの終了処理を行なう。

【0045】図8では割当て中の「仕業3」「仕業4」共に終了させる必要はないと判断し、次に割当てを行なうジョブがあると判断すると(ステップA13)、再度上記ステップA1からの処理を繰返す。

【0046】次に「運行11」をジョブデータベース11より取出し(ステップA1)、リソースデータベース12より「仕業3」を取出す(ステップA2)。この「運行11」を「仕業3」に接続不可と判断し(ステップA3)、まだリソースがあると判断して(ステップA6)、今度は「仕業4」を取出し(ステップA2)、こ

の「運行11」が「仕業4」に接続可能であるか否か判断する(ステップA3)。接続可能と判断し、次に割当てたものとして制約を満たすと判断すると(ステップA4)、この「運行11」を「仕業4」に割当てる(ステップA5)。割当てを終えた後、再び上記図8のリソースの終了処理を行なう。

【0047】図8では、割当て中の「仕業4」「仕業3」共に終了させる必要はないと判断し、次に割当てを行なうジョブがあると判断すると(ステップA13)、再度上記ステップA1からの処理を繰返す。

【0048】次いで「運行12」をジョブデータベース11より取出し(ステップA1)、「仕業3」を取出す(ステップA2)。この「運行12」を「仕業3」に接続可能と判断し(ステップA3)、次に割当てたものとして制約を満たすと判断すると(ステップA4)、この「運行12」を「仕業3」に割当てる(ステップA5)。割当てを終えた後、再び上記図8のリソースの終了処理を行なう。

【0049】図8では割当て中の「仕業3」「仕業4」共に終了させる必要はないと判断し、次に割当てを行なうジョブがあると判断すると(ステップA13)、再度上記ステップA1からの処理を繰返す。

【0050】次に「運行13」をジョブデータベース11より取出し(ステップA1)、「仕業3」を取出す(ステップA2)。この「運行13」を「仕業3」に接続可能であると判断した後(ステップA3)、「運行13」を「仕業3」に割当てたものとして「最大ハンドル時間」が「210分」となり、制約条件「200分」を超えてしまうこととなるので、この制約を満たさないと判断して(ステップA4)、まだ割当てを行なうリソースがあるか否か判断する(ステップA6)。ここではまだ「仕業4」が残っているので、上記ステップA2に進んで「仕業4」を取出し(ステップA2)、今度は「運行13」が「仕業4」に接続可能であるか否か判断する(ステップA3)。接続可能と判断し、次に割当てたものとして制約を満たすと判断すると(ステップA4)、この「運行13」を「仕業4」に割当てる(ステップA5)。割当てを終えた後、再び上記図8のリソースの終了処理を行なう。

【0051】図8では、割当て中の「仕業4」を取込み(ステップB1)、次に選択される筈のジョブをジョブデータベース11から取込む(ステップB2)。この場合、「運行13」が最後にジョブであり、次のジョブはないため、仮想的にジョブの開始時刻が「仕業4」の終了時刻目安を超えていると判断し(ステップB3)、この「仕業4」に関しては割当てを終了させるものとして終了設定を行ない(ステップB4)、次いでまだ割当てが終了していないリソースがあるか否か判断する(ステップB5)。ここではまだ「仕業3」に関して割当てが終了していないため、再び上記ステップB1からの処理



を繰返し、この「仕業3」に関しても同様に割当ての終了設定を行なう。以上でこの図8の処理を終え、割当てを行なっていないジョブがもうないと判断すると（ステップA13）、以上でこの図7による割当て処理もすべて終了する。

【0052】上記のように割当て処理を実行した結果、図9に示すように「仕業1」には「運行1」「運行2」「運行4」及び「運行6」、「仕業2」には「運行3」「運行5」及び「運行7」、「仕業3」には「運行8」「運行10」及び「運行12」、「仕業4」には「運行9」「運行11」及び「運行13」が割当てられることとなる。図10はこれを時間の流れで示したもので、上記図6に示した「知識4・時刻順に取出される次のジョブの開始時刻が、（リソースの）終了時刻目安を超えている時、該当するリソースへの割当てを終了する。」なる知識を用いることで、割当てする必要のないリソースに関しては割当て可能であるか否かの判断処理を省略することで、処理に要する時間を大幅に短縮できただけでなく、各「仕業」のハンドル時間がほぼ等しくなっており、また、すべての「仕業」が休憩をとっているため、割当て結果としては優れたものとなっていることがわかる。

【0053】これに対して、「仕業」の終了時刻に関する知識を利用しないで割当て処理を実行すると、図11に示すように「運行8」が「仕業1」「仕業2」に接続可能で、且つ「仕業2」については他の制約条件も満足してしまうためにこの「仕業2」に割当てられてしまうこととなる。その後、「運行9」～「運行13」に関しては「仕業1」「仕業2」が制約条件を満たさないとして「仕業3」「仕業4」に順次割当てることとなる。その結果、同図11に示すように「仕業1」「仕業2」及び「仕業3」と「仕業4」とが「拘束時間」「ハンドル時間」の点で不均衡となり、また、「仕業4」が休憩をとっていないこととなるため、割当て結果としては甚だ具合の悪いものとなる。

【0054】なお、上記実施例ではいずれも朝早い運行から時刻の順に割当てる例を示したが、反対に運行の時刻を遡って夜遅くから順に割当てるようにしても、「開始時刻目安」の制約条件を用いることで同様の結果が得られる。

【0055】

【発明の効果】以上詳記した如く本発明によれば、少な

【図9】

仕業1	午前	運行1、運行2、運行4、運行6
仕業2	午前	運行3、運行5、運行7
仕業3	午後	運行8、運行10、運行12
仕業4	午後	運行9、運行11、運行13

くとも開始時刻及び終了時刻の属性を有するジョブを複数記憶保存するジョブデータベースと、リソースを複数記憶保存するリソースデータベースと、ジョブをリソースに割当て際の絶対的レベルでの制約条件と開始時刻及び終了時刻の目安となる標準的レベルでの制約条件とをそれぞれ記憶保存する制約データベースと、上記ジョブデータベースに保存されるジョブをその時刻の順に取出して上記制約データベースに保存される絶対的レベルでの制約条件及び標準的レベルでの制約条件に従い、上記リソースデータベースに保存されるリソースに割当てする推論部と、上記制約データベースに保存される制約条件の内容を書換変更するマンマシンインタフェースとを備えるようにしたので、リソースの要求する事情に即してより詳細な制約条件をその制約の度合いに応じて設定し、リソースに適した優れた割当て決定を容易に行なうことが可能な割当て装置を提供できる。

【図面の簡単な説明】

【図1】本発明の一実施例に係る構成を示すブロック図。

【図2】図1のジョブデータベースに格納されるジョブを例示する図。

【図3】図1の制約データベースに格納される制約条件を例示する図。

【図4】図1の制約データベースに格納される制約条件を例示する図。

【図5】図1の制約データベースに格納される制約条件を例示する図。

【図6】図1の推定部のアルゴリズムの基準となる知識を例示する図。

【図7】図1の推定部による割当て決定の処理内容を示すフローチャート。

【図8】図7のリソースの終了処理のサブルーチンを示すフローチャート。

【図9】図7の処理によって得られる割当て結果を示す図。

【図10】図7の処理によって得られる割当て結果を時間に対応して示す図。

【図11】悪い割当て結果の例を示す図。

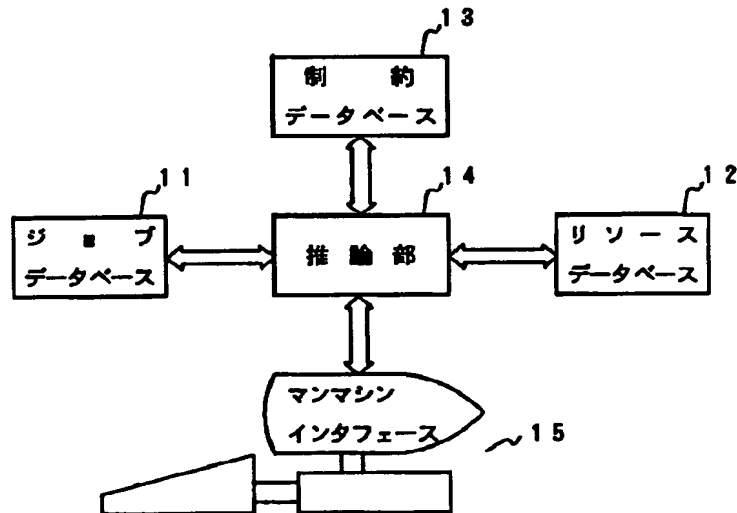
【符号の説明】

11…ジョブデータベース、12…リソースデータベース、13…制約データベース、14…推定部、15…マンマシンインタフェース。

【図10】

	0	7	8	9	10	11	12	13	14	15	16	17
仕業1	1	2		4	9							
仕業2			3	5		7						
仕業3							8		10	12		
仕業4								9	11	13		

【図1】



【図4】

から へ	午 前	午 後	日 動	中 休	午前増務	午後増務
午 前		10分				120分
午 後						
日 動						
中 休						
午前増務		120分				
午後増務						

【図11】

	6	7	8	9	10	11	12	13	14	15	16	17
仕事1	1	2		4	5							
仕事2			3	6		7	8					
仕事3								9	11		13	
仕事4									10	12		

【図2】

	出発場所	到着場所	出発時刻	到着時刻
運行1	東京	品川	6:00	7:00
運行2	品川	東京	7:10	8:10
運行3	東京	品川	8:20	9:20
運行4	東京	東京	9:10	9:40
運行5	品川	東京	9:30	10:30
運行6	東京	東京	9:50	10:20
運行7	東京	東京	11:30	12:00
運行8	東京	東京	12:10	12:40
運行9	東京	品川	12:45	13:45
運行10	東京	品川	13:40	14:40
運行11	品川	東京	13:55	14:55
運行12	品川	東京	14:50	15:50
運行13	東京	東京	15:55	15:55

【図3】

	午 前	午 後	日 動	中 休	午前増務	午後増務
最大拘束時間	270分	270分				
最大ハンドル時間	200分	200分				
連続ハンドル時間	130分	130分				
開始時刻目安	5:00	12:00				
終丁時刻目安	12:00	22:00				
休憩時間	50分	50分				

【図5】

停 留 所	東京 品川		
休 憩 場 所	東京		
連続ハンドル 中 断 場 所	東京 品川		
最小折待時間	東京	5分	0分～24時
	品川	5分	0分～24時

【図6】

知識1    ジョブが接続可能で、かつ接続した時、制約を満たすリソースがあるなら、該リソース中の配列順序の早いリソースに割当てる。

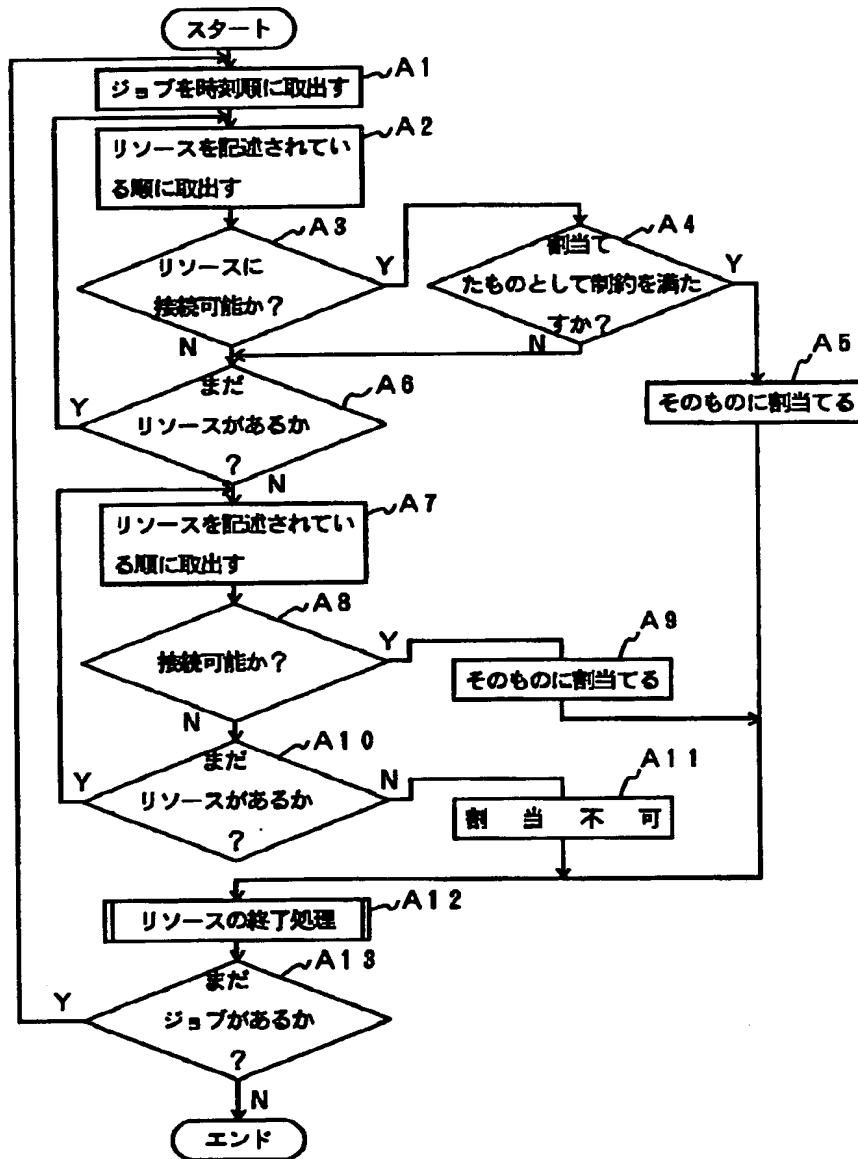
知識2    ジョブが接続可能な時、配列順序の早いリソースに割当てる。

知識3    どのリソースにも接続不可能な時、割当不可とする。

知識4    時刻順に取出される次のジョブの開始時刻が、終了時刻目安を超えている時、該当するリソースへの割当てを終了する。

知識5    全てのジョブが割当てもしくは割当不可とされた時、割当て中であるリソースへの割当てを終了する。

【図7】



【図8】

